

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №4**

з дисципліни  
«Дискретна математика»

**Виконав:**

студент групи КН-109  
Гречух Тарас

**Викладач:**

Мельникова Н.І.

Львів – 2018 р.

**Тема:** Основні операції над графами. Знаходження оєства мінімальної ваги за алгоритмом Пріма-Краскала.

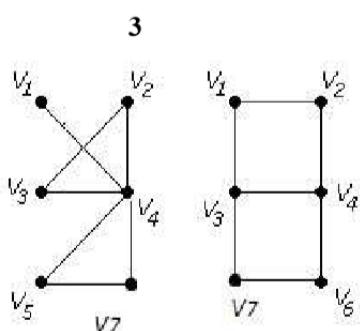
**Мета:** Набуття практичних вмінь та навичок з використання алгоритмів Пріма і Краскала.

**Завдання з додатку 1 (варіант 3):**

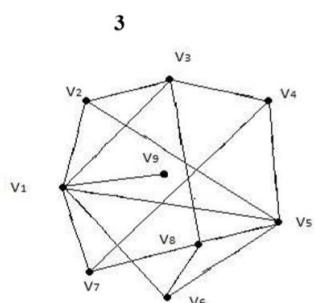
Розв'язати на графах наступні задачі:

1. Виконати наступні операції над графами:

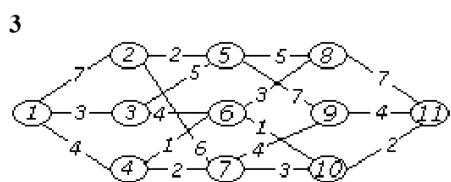
- 1) знайти доповнення до першого графу,
- 2) об'єднання графів,
- 3) кільцеву суму  $G_1$  та  $G_2$  ( $G_1+G_2$ ),
- 4) розщепити вершину у другому графі,
- 5) виділити підграф А, що складається з 3-х вершин в  $G_1$  і знайти стягнення А в  $G_1 \setminus A$ ,
- 6) добуток графів



2. Знайти таблицю суміжності та діаметр графа.



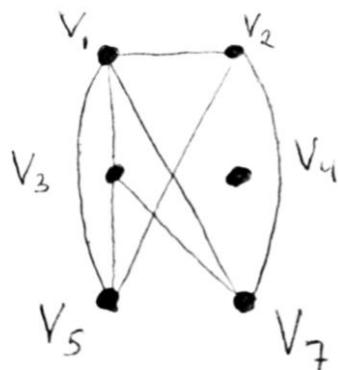
3. Знайти двома методами (Краскала і Прима) мінімальне оствове дерево графа.



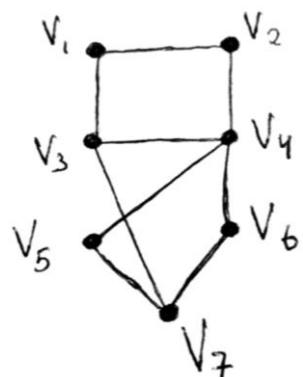
**Розв'язання:**

1)

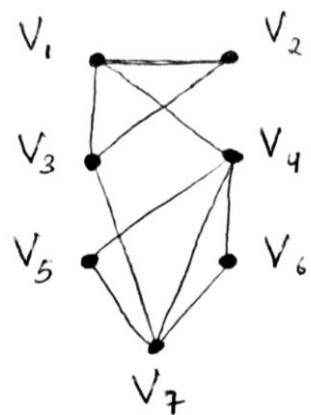
1. Доповнення до першого графу



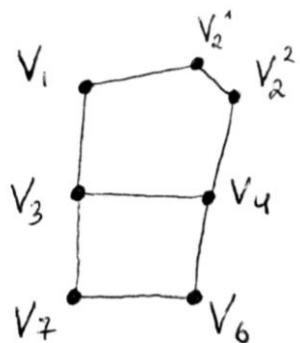
2. Об'єднання графів



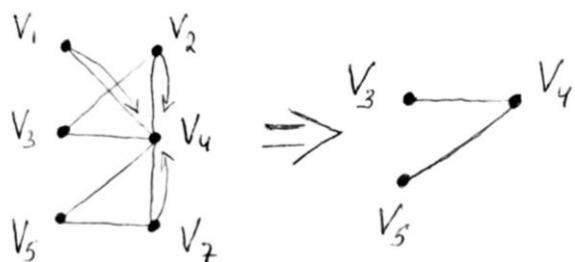
3. Кільцева сума графів



4. Розщепити вершину у другому графі

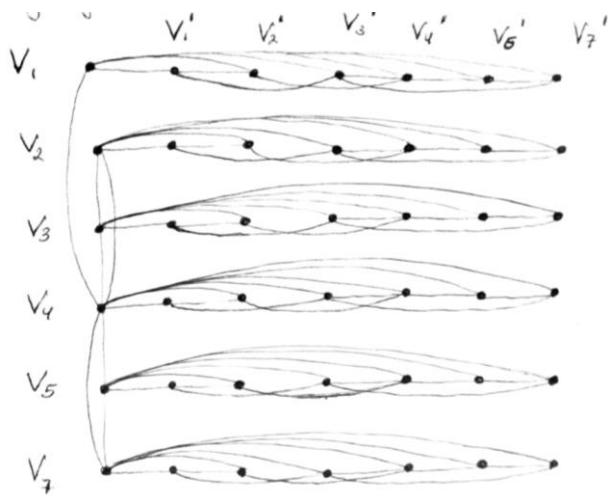


5. Виділення підграфу A, що складається з 3-х вершин в G1 і знаходження стягнення A в G1 ( $G1 \setminus A$ ),



Ребро V1-V4 стягується у вершину V4, ребро V2-V4 теж у вершину V4 і ребро V7-V4 аналогічно у вершину V4.

6.



2)

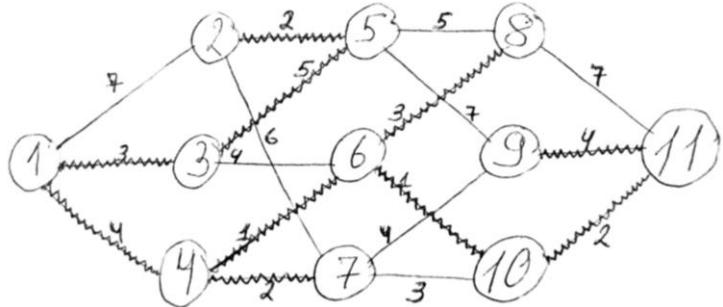
Таблиця (матриця) суміжності:

	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	0	1	1	0	1	1	1	0	1
V2	1	0	1	0	1	0	0	0	0
V3	1	1	0	1	0	0	0	1	0
V4	0	0	1	0	1	0	1	0	0
V5	1	1	0	1	0	1	0	1	0
V6	1	0	0	0	1	0	0	1	0
V7	1	0	0	1	0	0	0	1	0
V8	0	0	1	0	1	1	1	0	0
V9	1	0	0	0	0	0	0	0	0

Діаметр графа –  $D = 3$ .

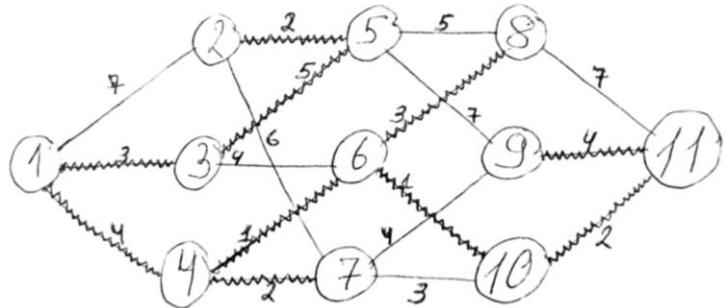
3)

(1) Метод Крускала:



Послідовність дій: Замальовуємо найменші ребра графа так, щоб не утворювалося циклу (4-6, 6-10, 4-7, 10-11, 2-5, 1-3, 6-8, 1-4, 9-11, 3-5).

(2) Метод Прима:



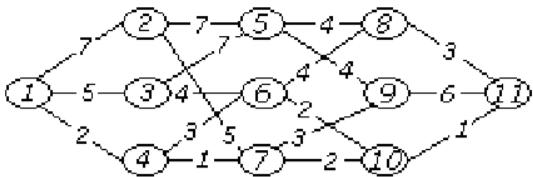
Послідовність дій: Обираємо довільну вершину і вже від неї поступово замальовуємо найменші ребра, але лише ті, що уже належать нашому шляху. Також уникаємо циклу. (4-6, 6-10, 4-7, 10-11, 6-8, 9-11, 1-4, 1-3, 3-5, 2-5).

### Завдання з додатку 2 (варіант 3):

Написати програму, яка реалізує алгоритм знаходження оствового дерева мінімальної ваги згідно свого варіанту.

**Варіант № 3**

За алгоритмом Прима знайти мінімальне остове дерево графа. Етапи розв'язання задачі виводити на екран. Протестувати розроблену програму на наступному графі:



### Розв'язання:

```
#include <iostream>
#include <vector>
#include <set>

using namespace std;

int main(void)
{
    vector< vector<int> > graph = {
        { 0, 7, 5, 2, 0, 0, 0, 0, 0, 0, 0 },
        { 7, 0, 0, 0, 7, 0, 5, 0, 0, 0, 0 },
        { 5, 0, 0, 0, 7, 4, 0, 0, 0, 0, 0 },
        { 2, 0, 0, 0, 0, 3, 1, 0, 0, 0, 0 },
        { 0, 7, 7, 0, 0, 0, 0, 4, 4, 0, 0 },
        { 0, 0, 4, 3, 0, 0, 0, 4, 0, 2, 0 },
        { 0, 5, 0, 1, 0, 0, 0, 0, 3, 2, 0 },
        { 0, 0, 0, 0, 4, 4, 0, 0, 0, 0, 3 },
        { 0, 0, 0, 0, 4, 0, 3, 0, 0, 0, 6 },
        { 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 1 },
        { 0, 0, 0, 0, 0, 0, 3, 6, 1, 0 }
    };
    set<int> SP = {}, SM = { 0,1,2,3,4,5,6,7,8,9,10 };

    int min = numeric_limits<int>::max();
    int l, t;

    for (auto i = 0; i < graph.size(); ++i)
        for (auto j = i; j < graph.size(); ++j)
            if (graph[i][j] < min && graph[i][j] != 0)
            {
                min = graph[i][j];
                l = i;
                t = j;
            }
    cout << "[" << l + 1 << ", " << t + 1 << " ];" << endl;
    SP.insert(l); SM.erase(l);
    SP.insert(t); SM.erase(t);
    while (!SM.empty())
    {
        min = numeric_limits<int>::max();
        l = t = 0;
        for (auto i : SM)
```

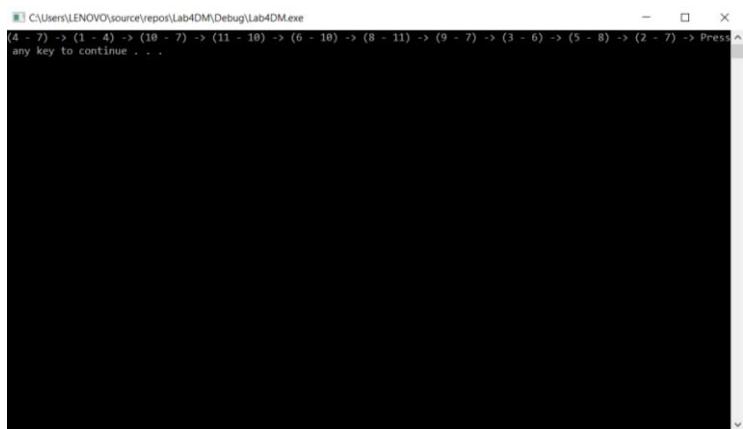
```

        for (auto j : SP)
            if (graph[i][j] < min && graph[i][j] != 0)
            {
                min = graph[i][j];
                l = i;
                t = j;
            }
        SP.insert(l);    SM.erase(l);
        cout << "[" << l + 1 << ", " << t + 1 << " ];" << endl;
    }

    system("pause");
}

```

### Результат виконаної програми:



**Висновок:** Я набув практичних вмінь та навичок з використання алгоритмів Пріма і Краскала.